



Programas

Facultad de Ingeniería

Análisis Numérico

Nombre: Herrera Godina Diana Celeste

Número de cuenta: 316161927

Grupo: 15

Profesor: Flores Delgado Gerardo

Ejecución Programa Derivación (centrada):

```
-----{ MENU DERIVADAS }-----  
1. Derivada Centrada  
2. Derivada Regresiva  
3. Derivada Progresiva  
4. Salir  
-----  
Ingrese la opcion que desee:      1  
Valor inicial:3  
Valor final: 4  
Valor aproximar X0: 5  
Valor de h: 6  
---{ Tabulando }---  
3.0000  3.3476  
-----{ ORDEN DERIVADAS }-----  
1. 2 Puntos  
2. 3 Puntos  
-----  
Ingrese la opcion que desee:      1  
-----{ RESULTADO DERIVADAS }-----  
1a. derivada centrada: 42.299327  
2a. derivada centrada: 13.714776  
-----
```

Ejecución Programa Derivación (regresiva):

```
-----{ MENU DERIVADAS }-----  
1. Derivada Centrada  
2. Derivada Regresiva  
3. Derivada Progresiva  
4. Salir  
-----  
Ingrese la opcion que desee:      2  
Valor inicial:3  
Valor final: 4  
Valor aproximar X0: 5  
Valor de h: 6  
---{ Tabulando }---  
3.00000  3.3476  
-----{ ORDEN DE DERIVADA }-----  
1. 2 Puntos  
2. 3 Puntos  
-----  
Ingrese la opcion que desee:      2  
-----{ RESULTADO DERIVADAS }-----  
1a. derivada regresiva orden2: 1.747828  
2a. derivada regresiva orden2: 0.400330  
-----
```

Ejecución Programa Derivación (progresiva):

```
-----{ MENU DERIVADAS }-----  
1. Derivada Centrada  
2. Derivada Regresiva  
3. Derivada Progresiva  
4. Salir  
-----  
Ingrese la opcion que desee:      3  
Valor inicial: 4  
Valor final: 6  
Valor aproximar X0: 4  
Valor de h: 8  
---{ Tabulando }---  
4.0000  4.1999  
-----{ ORDEN DERIVADAS }-----  
1. 2 Puntos  
2. 3 Puntos  
-----  
Ingrese la opcion que desee:      2  
-----{ RESULTADO DERIVADAS }-----  
1a. derivada Progresiva 3 Puntos: -76092.126264  
2a. derivada Progresiva 3 Puntos: -28858110.430413  
-----  
-----{ MENU DERIVADAS }-----  
1. Derivada Centrada  
2. Derivada Regresiva  
3. Derivada Progresiva  
4. Salir  
-----  
Ingrese la opcion que desee:      -
```

Programa Integración Ejecución:

```
---{ Programa Integracion }---  
Limite inferior: 2  
Limite superior: 4  
Numero de particiones: 5  
----{ Valores de Y }----  
Ingrese el valor valor de y:      6  
----{ Valores de Y }----  
Ingrese el valor valor de y:      5  
----{ Valores de Y }----  
Ingrese el valor valor de y:      6  
----{ Valores de Y }----  
Ingrese el valor valor de y:      7  
----{ Valores de Y }----  
  
Ingrese el valor valor de y:      6  
2.0000  29.5562  
2.4000  63.4935  
2.8000  128.9260  
3.2000  251.2131  
3.6000  474.3131  
4.0000  873.5704  
  
La integral aprox. Trapecio= 358.078379  
Repetir s/n: _
```

Programa Derivación (código)

```
//Herrera Godina Diana Celeste
#include <stdio.h>
#include <math.h>
#define DIM 20

double x[DIM],y[DIM];

double f (double x){
    return exp (x)/(pow (x,2)-3);
}

void tabula (double xi,double xf,double deltaX) {
    int i=0;

    while (xi <= xf) {
        x[i] = xi;
        y[i] = f (xi);
        printf ("%4lf\t%4lf\n",x[i],y[i]);
        xi += deltaX;
        i++;
    }
}

//primera derivada dos puntos
double primeraDerivadaProgresiva (double x, double h) {
    return (f(x+h) - f (x))/h;
}

double primeraDerivadaCentrada (double x, double h) {
    return (f(x+h) - f (x-h))/(2*h);
}

double primeraDerivadaRegresiva (double x, double h) {
    return (f(x) - f (x-h))/(h);
}

//segunda derivada dos puntos
double segundaDerivadaProgresiva (double x, double h) {
    return (f(x)-2*f(x+h)+f(x+2*h))/(h*h);
}

double segundaDerivadaCentrada (double x, double h) {
    return (f(x-h)-2*f(x)+f(x+h))/(h*h); // [1 -2 1]
}

double segundaDerivadaRegresiva (double x, double h) {
    return (f(x-2*h)-2*f(x-h)+f(x))/(h*h);
}

//Primera derivada 3 puntos
double primeraDerivada3puntos (double x, double h){
    return (-3*f(x)+4*f(x+h)-f(x+2*h))/(2*h);
}

double primeraDerivadaCentradaOrden4 (double x, double h){
    return (f(x-2*h)-8*f(x-h)+8*f(x+h)-f(x+2*h))/(12*h);
}

double primeraDerivadaRegresivaSegundoOrden (double x, double h) {
    return (f(x-2*h)-4*f(x-h)+3*f(x))/(2*h);
}
```

```

//Segunda derivada 3 puntos
double segundaDerivadaProgresivaOrden2 (double x, double h) {
    return (2*f(x)-5*f(x+h)+4*f(x+2*h)-f(x+3*h))/(h*h);
}
double segundaDerivadaCentradaOrden4(double x, double h) {
    return (-f(x-2*h)+16*f(x-h)-30*f(x)+16*f(x+h)-f(x+2*h))/(12*h*h);
}
double segundaDerivadaRegresivaOrden2(double x, double h) {
    return (-f(x-3*h)+4*f(x-2*h)-5*f(x-h)+2*f(x))/(h*h);
}

int main () {
    int c, op, centrada, regresiva, progresiva;
    double x,xi,xf,deltaX;

    do {
        printf("\n ----{ MENU DERIVADAS }----\n");
        printf("\n 1. Derivada Centrada");
        printf("\n 2. Derivada Regresiva");
        printf("\n 3. Derivada Progresiva");
        printf("\n 4. Salir ");
        printf("\n-----");
        printf("\n\nIngrese la opcion que desee:\t");
        scanf( "%d", &op );

        switch ( op )
        {
            case 1:
                printf ("\nValor inicial:");
                scanf ("%lf", &xi);
                printf ("\nValor final: ");
                scanf ("%lf", &xf);
                printf ("\nValor aproximar X0: ");
                scanf ("%lf", &x);
                printf ("\nValor de h: ");
                scanf ("%lf", &deltaX);
                printf("\n---{ Tabulando }---\n");
                tabula(xi,xf,deltaX);

                printf("\n----{ ORDEN DERIVADAS }----\n");
                printf( "\n 1. 2 Puntos");
                printf( "\n 2. 3 Puntos\n");
                printf("\n-----");
                printf(" \nIngrese la opcion que desee:\t");
                scanf( "%d", &centrada );

                switch ( centrada ) {
                    case 1:
                        printf("\n----{ RESULTADO DERIVADAS }-----\n");
                        printf ("\n1a. derivada centrada: %lf\n",primeraDerivadaCentrada(x,deltaX));
                        printf ("\n2a. derivada centrada: %lf\n",segundaDerivadaCentrada(x,deltaX));
                        printf("\n-----\n");
                        break;

                    case 2:
                        printf("\n----{ RESULTADO DERIVADAS }-----\n");

```

```

printf("\n1a.derivadacentradaorden4:%lf\n",primeraDerivadaCentradaOrden4(x,deltaX));
printf ("\n2a. derivada centrada orden4: %lf\n",segundaDerivadaCentradaOrden4(x,deltaX));
printf("\n-----\n");
break;
}

break;

case 2:
    printf ("\nValor inicial:");
    scanf ("%lf", &xi);
    printf ("\nValor final: ");
    scanf ("%lf", &xf);
    printf ("\nValor aproximar X0: ");
    scanf ("%lf", &x);
    printf ("\nValor de h: ");
    scanf ("%lf", &deltaX);
    printf("\n---{ Tabulando }---\n");
    tabula(xi,xf,deltaX);

printf("\n---{ ORDEN DE DERIVADA }---\n");
printf("\n 1. 2 Puntos");
    printf("\n 2. 3 Puntos ");
    printf("\n-----");
    printf("\nIngresa la opcion que desee: \t");
    scanf( "%d", &regresiva );

    switch ( regresiva ) {

        case 1:
            printf("\n---{ RESULTADO DERIVADAS }-----\n");
            printf("\n1a.          derivada          regresiva:
%lf\n",primeraDerivadaRegresiva(x,deltaX));
            printf          ("\n2a.          derivada          regresiva:
%lf\n",segundaDerivadaRegresiva(x,deltaX));
            printf("\n-----\n");
            break;

        case 2:
            printf("\n---{ RESULTADO DERIVADAS }-----\n");
            printf          ("\n1a.          derivada          regresiva          orden2:
%lf\n",primeraDerivadaRegresivaSegundoOrden(x,deltaX));
            printf          ("\n2a.          derivada          regresiva          orden2:
%lf\n",segundaDerivadaRegresivaOrden2(x,deltaX));
            printf("\n-----\n");
            break;
    }
    break;

case 3:
    printf ("\nValor inicial: ");
    scanf ("%lf", &xi);
    printf ("\nValor final: ");
    scanf ("%lf", &xf);
    printf ("\nValor aproximar X0: ");
    scanf ("%lf", &x);
    printf ("\nValor de h: ");

```



```

scanf ("%lf", &deltaX);
printf("\n---{ Tabulando }---\n");
tabula(xi,xf,deltaX);

printf("\n----{ ORDEN DERIVADAS }----\n");
printf("\n 1. 2 Puntos");
printf("\n 2. 3 Puntos");
printf("\n-----");
printf("\nIngrese la opcion que desee: \t");
scanf("%d", &progresiva);

switch ( progresiva ) {

    case 1:
        printf("\n----{ RESULTADO DERIVADAS }-----\n");
        printf (" \n1a. derivada Progresiva 2 puntos:

%lf\n",primeraDerivadaProgresiva(x,deltaX));
        printf (" \n2a. derivada Progresiva 2 puntos:

%lf\n",segundaDerivadaProgresiva(x,deltaX));
        printf("\n-----\n");
        break;

    case 2:
        printf("\n----{ RESULTADO DERIVADAS }-----\n");
        printf (" \n1a. derivada Progresiva 3 Puntos:

%lf\n",primeraDerivada3puntos(x,deltaX));
        printf (" \n2a. derivada Progresiva 3 Puntos:

%lf\n",segundaDerivadaProgresivaOrden2(x,deltaX));
        printf("\n-----\n");
        break;

    }

    break;

}

fflush (stdin); // Limpia el buffer del teclado

} while (op != 4);
}

```

Programa Integración (código)

```
//Herrera Godina Diana Celeste
#include <stdio.h>
#include <math.h>
#define DIM 100
double y[DIM],x[DIM];

void tabula (double xi,double xf,double deltaX) {
    int i=0;

    while (xi <= xf) {
        x[i] = xi;
        y[i] = a+h*i;
        printf ("%4lf\t%4lf\n",x[i],y[i]);
        xi += deltaX;
        i++;
    }
    printf ("\n");
}

double Trapecio (double a, double b, int n) {
    int i;
    double h,integral,suma;

    h= (b-a)/n;
    tabula(a,b,h);
    suma = 0;
    for (i = 1; i < n-1; i++)
        suma= suma + y[i];
    integral = h/2.0 * (y[0] + 2*suma + y[1] );
    return integral;
}
```

```

double S13 (double a, double b, int n) {
    int i;
    double h,integral,sumaP,sumal;

    h= (b-a)/n;
    tabula(a,b,h);
    sumaP = 0;
    for (i = 2; i <= n-1; i=i+2)
        sumaP= sumaP + y[i];
    sumal = 0;
    for (i = 1; i <= n-1; i=i+2)
        sumal= sumal + y[i];
    printf ("h= %lf, sumaP= %lf, sumal= %lf\n",h,sumaP,sumal);
    integral = h/3.0 * (y[0] + 2*sumaP + 4*sumal+ y[2]));
    return integral;
}

double S38 (double a, double b, int n) {
    int i;
    double h,integral,sumaM3,sumaNoM3;

    h= (b-a)/n;
    tabula(a,b,h);
    sumaM3 = 0;
    for (i = 3; i <= n-1; i=i+3)
        sumaM3= sumaM3 + y[i];
    sumaNoM3 = 0;
    for (i = 1; i <= n-1; i++)
        if (i % 3)
            sumaNoM3= sumaNoM3 + f (a+h*i);
    printf ("h= %lf, sumaM3= %lf, sumaNoM3= %lf\n",h,sumaM3,sumaNoM3);
    integral = 3*h/8.0 * (y[0]+ 3*sumaNoM3 + 2*sumaM3+ y[2]);
    return integral;
}

main () {

```

```

int d,i,c,n;
double x,xi,xf,deltaX,a,b;

do {
    fflush (stdin);
    printf("\n---{ Programa Integracion }---\n");
    printf ("\nLimite inferior: ");
    scanf ("%lf",&a);
    printf ("\nLimite superior: ");
    scanf ("%lf",&b);
    printf ("\nNumero de particiones: ");
    scanf ("%d",&n);
    for(i=0;i<n;i++){
        printf("\n----{ Valores de Y }----\n");
        printf("\nIngrese el valor valor de y:\t");
        scanf("%d",&y[i]);
    }
    if (n % 3 == 0) // Multiplo de 3
        printf ("\nLa integral aprox. 3/8= %lf\n", S38 (a,b,n));
    else if (n % 2 == 0) // Multiplo de 2
        printf ("\nLa integral aprox. 1/3= %lf\n", S13 (a,b,n));
    else
        printf ("\nLa integral aprox. Trapecio= %lf\n", Trapecio (a,b,n, y[i]));
    printf ("Repetir s/n: ");
    fflush (stdin); // Limpia el buffer del teclado
    c= getchar ();
} while (c == 's');

}

```